IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Applicant:	Grover et al.	Patent Application	
Application No.:	10/621,207	Group Art Unit:	2192
Filed:	July 15, 2003	Examiner:	Dao, Thuy Chan

For: HANDLING EXCEPTIONS

APPEAL BRIEF

Table of Contents

I. Real Party in Interest

The assignee of the present application is Hewlett-Packard Development Company, L.P.

II. Related Appeals and Interferences

There are no related appeals or interferences known to the Appellants.

III. Status of Claims

Claims 1-8, 10-15 and 17 remain pending. Claims 1-8, 10-15 and 17 are rejected. Claims 9 and 16 are cancelled. This Appeal involves Claims 1-8, 10-15 and 17.

IV. Status of Amendments

No amendment subsequent to the Final Action has been filed in this case.

V. Summary of Claimed Subject Matter-

Independent Claims 1, 10 and 15 of the present application pertain to embodiments associated with handling exceptions.

Independent Claim 1 recites "[a]n exception handling mechanism stored in one or more computer-readable storage devices." This embodiment is depicted at least in Figure 2. The exception handling mechanism comprises an exception handler (210 of Figure 2) for recording exception information dependant on types of exceptions and programming tasks that encounter exceptions (page 7, lines 13-15). The exception handling mechanism further comprises a recovery agent (220 of Figure 2) for taking an action upon an occurrence of an exception that occurred for a programming task (page 7, lines 5-7), wherein the action is performed outside of a debugging operation (page 7, lines 12-13). The exception handling mechanism further comprises wherein the action to be taken upon the occurrence of the exception corresponds to a type of exception and a programming task, and includes one or a combination of restarting the programming task, terminating the programming task, resetting a system running the programming task, and disregarding the exception (page 9, lines 7-21). The exception handling mechanism further comprises wherein the exception handler and the recovery agent run on a first system that operates autonomously and the first system is embedded in a second system (page 6, lines 11-18).

Independent Claim 10 recites "[a] processing system stored in one or more computer-readable storage devices." This embodiment is depicted at least in Figures 1, 2 and 4. The processing system comprises a first system (110 of Figure 1 and page 5, lines 9-11). The processing system further comprises an autonomous second system embedded in the first system (120 of Figure 1; page 5, lines 9-11; and page 6 lines 11-18). The processing system further comprises an exception handler (210 of Figure 2) running in the second system for recording exception information upon an occurrence of an exception in the second system (page 7, lines 13-15 and page 6 lines 11-18). The processing system further comprises a recovery agent (220 of Figure 2) running on the second system (page 6 lines 11-18), for taking an action upon the occurrence of the exception based on the recorded exception information (page 9, lines 7-11), wherein the action is performed outside of a debugging operation (page 7, lines 12-13). The processing system further comprises wherein the action corresponds to a type of exception that occurred in a programming task (page 7, lines 5-7).

Independent Claim 15 recites "[a] computer-readable storage device having stored thereon a computing system" This embodiment is depicted at least in Figures 1, 2 and 4. The computer readable storage device further comprising an exception handler (210 of Figure 2) for recording exception information on non-volatile memory upon an occurrence of an exception (page 7 lines 19-20). The computer readable storage device further comprising a recovery agent (220 of Figure 2) for taking an action upon the occurrence of the exception based on the recorded exception information (page 9, lines 7-11), wherein the action is performed outside of a debugging operation (page 7, lines 12-13). The computer readable storage device further comprising an analysis tool for identifying the cause of the exception (230 of Figure 2 and page 6 lines 23-24). The computer readable storage device further comprising wherein the analysis tool receives the exception information from the nonvolatile memory via an interface interfacing a first system and a second system running the exception handler and the recovery agent wherein the second system is embedded in a third system and the second system operates autonomously of other systems (page 10 line 14 – page 11 line 3; and page 6, lines 11-18).

VI. Grounds of Rejection to Be Reviewed on Appeal

1. Claims 1-8, 10-15 and 17 are rejected under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 6,339,832 to Bowman-Amuah.

VII. Argument

1. Whether Claims 1-8, 10-15 and 17 are rejected under 35 U.S.C. §102(b) as being anticipated by Bowman-Amuah.

Appellants have reviewed Bowman-Amuah and respectfully submit that the embodiments recited in Claims 1-8, 10-15 and 17 are not anticipated by Bowman-Amuah for at least the following rationale.

A. Bowman-Amuah does not disclose every claimed element.

Appellants respectfully point out that Claim 1 recites (Claims 10 and 15 include similar features):

> An exception handling mechanism stored in one or more computer-readable storage devices, said exception handling mechanism comprising:
>
> an exception handler for recording exception information dependant on types of exceptions and programming tasks that encounter exceptions; and
>
> a recovery agent for taking an action upon an occurrence of an exception that occurred for a programming task, wherein the action is performed outside of a debugging operation;
>
> wherein the action to be taken upon the occurrence of the exception corresponds to a type of exception and a programming task, and includes one or a combination of restarting the programming task, terminating the programming task, resetting a system running the programming task, and disregarding the exception,
>
> wherein the exception handler and the recovery agent run on a first system that operates autonomously and the first system is embedded in a second system."

MPEP §2131 provides:

> "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). … "The identical invention must be shown in as complete detail as is contained in the ... claim." *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). The elements must be arranged as required by the claim.

The final Office Action mailed January 29, 2009 (hereinafter referred to as "instant Office Action) states that "Claim 1: Bowman-Amuah discloses an exception handling mechanism stored in one ore [sic] more computer-readable storage devices comprising: an exception handler for recording exception information (e.g., FIG. 145, exception response table, co1.262: 61 - co1.263: 67; co1.264: 45 - co1.265: 33) dependant on types of exceptions and programming tasks that encounter exceptions (e.g., FIG. 143, sheet 87, each type of exception A-C has each handling logic A-C, co1.260: 54 - co1.261: 37; co1.263: 28-67; co1.264: 9-44; co1.265: 61 co1.266: 56); and a recovery agent for taking an action upon an occurrence of an exception that occurred for a programming task (e.g., FIG. 55, programming classes and subclasses of batch jobs (programming task), co1.193: 41 - co1.194: 34; co1.93: 24-50), wherein the action is performed outside of a debugging operation (e.g., FIG.10, Base Services with Batch jobs, co1.31: 57 - co1.32; 38; FIG. 28, Batch jobs exceptions are handled outside a debugging operation, col. 106: 65 - col.109: 34)..." (emphasis added; instant Office Action page 4 line 23 – page 5 line 12).

Appellants respectfully submit that Bowman-Amuah does not anticipate "a recovery agent for taking an action upon an occurrence of an exception that occurred for a programming task, wherein the action is performed outside of a debugging operation" (emphasis added; Appellants' Claim 1). Appellants understand Bowman-Amuah to disclose "A system, method and article of manufacture are provided for recording exception handling requirements for maintaining a consistent error handling approach. An exception response table is provided in which an exception is recorded. The context of the exception is entered in the exception response table and a response for the exception is listed in the exception response table. The response is subsequently outputted upon the exception occurring in the context." (emphasis added; Bowman-Amuah Summary of the Invention). In other words, Bowman-Amuah discloses an exception handler that places exceptions in context according to what type of exception each exception is.

As stated above, the instant Office Action states, "a recovery agent for taking an action upon an occurrence of an exception that occurred for a programming task (e.g., FIG. 55, programming classes and subclasses of batch jobs (programming task), co1.193: 41 - co1.194: 34; co1.93: 24-50)." Appellants understand col. 193-4 to refer to FIG. 55 which does not disclose exceptions and exception handling, but instead discloses, "FIG. 55 illustrates a flowchart for a method 5500 for representing a plurality of batch jobs of a system each with a unique class" (emphasis added, Bowman-Amuah Col.193:24-45). Appellants respectfully submit that a "batch job", as disclosed in Bowman-Amuah, is not an exception and does not imply an exception or exception handling. Appellants respectfully point out that Bowman-Amuah refers to exceptions and exception handing in other portions of Bowman-Amuah and would have not have used the phrase "batch job" to refer to an exception.

Similarly, col.93:24-50 do not disclose "a recovery agent for taking an action upon an occurrence of an exception that occurred for a programming task" but instead discloses "[t]he automatic restart/recovery feature helps a system recognize when components have failed and attempts to restart them." (Bowman-Amuah, col.93:35-37). Appellants submit that this portion of Bowman-Amuah discloses recovery of "components" when "components have failed" and not what is disclosed in Appellants' Claim 1. Therefore Appellants submit that Bowman-Amuah does not disclose "a recovery agent for taking an action upon an occurrence of an exception that occurred for a programming task" (Appellants' Claim 1).

As stated above, the instant Office Action state, “wherein the action is performed outside of a debugging operation (e.g., FIG.10, Base Services with Batch jobs, co1.31: 57 - co1.32; 38; FIG. 28, Batch jobs exceptions are handled outside a debugging operation, col. 106: 65 - col.109: 34)” (emphasis added). Appellants respectfully submit that Bowman-Amuah does not disclose “Batch jobs exceptions are handled outside a debugging operation” as stated by the instant Office Action, nor does Bowman-Amuah disclose “Batch jobs exceptions”. Appellants understand Bowman-Amuah Fig.10 col.31:57-col.32:38 to disclose a “Framework Overview” for the “Netcentric Architecture Framework” and does not disclose “wherein the action is performed outside of a debugging operation” nor does it disclose the handling of exceptions.

Additionally, Bowman-Amuah discloses, “Batch processing is used to perform large scale repetitive processing where no user involvement is required as well as reporting. Areas for design attention include scheduling, recovery/restart, use of job streams and high availability (e.g. 24 hour running).” (emphasis added, Bowman-Amuah, col. 108 lines 37-41) “Batch application programs can include business processing such payroll, billing, etc. and can also include report generation.” (emphasis added, Bowman-Amuah, col. 108 lines 52-55). Appellants respectfully submit that “Batch processing” “used to perform large scale repetitive processing” and “Batch application programs” including “report generation,” as disclosed by Bowman-Amuah is not “wherein the action is performed outside of a debugging operation” nor does it disclose the handling of exceptions.

Additionally, Appellants understand FIG.28 col.106:65-col.109:34 to disclose descriptions of various components used by the technology in Bowman-Amuah and “[b]atch processing is used to perform large scale repetitive processing where no user involvement is required as well as reporting." Appellants submit that this portion of Bowman-Amuah does not refer to exceptions and exception handling nor does it refer to taking action regarding an exception outside of a debugging operation as disclosed in Appellants' Claim 1. As argued above, had this portion of Bowman-Amuah been referring to exceptions and exception handling, it would have used those terms as opposed to terms such as "batch jobs." Therefore, Appellants submit that Bowman-Amuah does not teach "a recovery agent for taking an action upon an occurrence of an exception that occurred for a programming task, wherein the action is performed outside of a debugging operation" (Appellants' Claim 1).

Claims 10 and 15 have features similar to Claim 1 and the same portions of Bowman-Amuah are relied upon in the instant Office Action. Therefore the above arguments apply to Claims 1, 10 and 15. Claims 2-8 depend from independent Claim 1, Claims 11-14 depend from independent Claim 10 and Claim 17 depends from independent Claim 15. Therefore the above arguments also apply to Claims 2-8, 11-14 and 17.

B. Bowman-Amuah does not disclose the claimed elements as arranged by Appellants' claims.

Appellants respectfully submit that the different portions of Bowman-Amuah cited by the instant Office Action do not disclose the claimed embodiments as arranged by the Appellants' Claims. As argued above, Appellants' do not concede that Bowman-Amuah discloses Appellants' claimed embodiments. However, Appellants submit that if the cited portions of Bowman-Amuah disclosed elements of the claimed embodiments, it does not disclose the embodiments as arranged by Appellants' Claims.

For example, the instant Office Action states, "Claim 1: Bowman-Amuah discloses an exception handling mechanism stored in one ore [sic] more computer-readable storage devices comprising: an exception handler for recording exception information (e.g., FIG. 145, exception response table, co1.262: 61 - co1.263: 67; co1.264: 45 - co1.265: 33) (the instant Office Action page 5 section 9). "[A]n exception handler for recording exception information" is the first element of Claim 1. The instant Office Action then states, "dependant on types of exceptions and programming tasks that encounter exceptions (e.g., FIG. 143, sheet 87, each type of exception A-C has each handling logic A-C, co1.260: 54 - co1.261: 37; co1.263: 28-67; co1.264: 9-44; co1.265: 61 co1.266: 56); and a recovery agent for taking an action upon an occurrence of an exception that occurred for a programming task (e.g., FIG. 55, programming classes and subclasses of batch jobs (programming task), co1.193: 41 - co1.194: 34; co1.93: 24-50), wherein the action is performed outside of a debugging operation (e.g., FIG.10, Base Services with Batch jobs, co1.31: 57 - co1.32; 38; FIG. 28, Batch jobs exceptions are handled outside a debugging operation, col. 106: 65 - col.109: 34)..." (emphasis added; instant Office Action page 4 line 23 – page 5 line 12). This references the second element of Appellants' Claim 1. In this portion of the instant Office Action, the columns cited in Bowman-Amuah include columns 31-32, 93, 106-109, 193-194, and 260-265.

On further inspection of the cited portions, the instant Office Action cites portions of Bowman-Amuah that disclose different embodiments of the invention disclosed in Bowman-Amuah. While columns 260-265 discuss exceptions and exceptions handling, columns 106-109 and 193-194 discuss batch jobs. As argued above, Appellants submit that exceptions and exceptions handling are not batch jobs. While Claim 1 is cited as an example, Claims 10 and 15 have features similar to Claim 1 and the instant Office Action relies on the same portions of Bowman-Amuah for the similar features. Therefore, because the different portions of Bowman-Amuah cited by the instant Office Action are separated by numerous pages and disclose different embodiments of the invention, Appellants submit that Bowman-Amuah does not disclose the embodiments of the Appellants' invention as arranged by the Claims.

C. Response to Arguments in the instant Office Action.

The instant Office Action states:

> "As an initial matter, the examiner notes that Applicants' arguments (in the cited paragraph above and all other locations) did not direct to the ground of rejection. As clearly set forth in page 5 of the previous Office action mailed August 13, 2008, the claimed limitation "an exception that occurred for a programming task" was equated with "Batch jobs exceptions", but not "batch job" as argued by the Applicants" (emphasis added, instant Office Action, page 2 lines 21-25).
>
> "Accordingly, Applicants' arguments fail to comply with 37 CFR 1.111(b). The reply by the applicant or patent owner did not distinctly and specifically points out the supposed errors in the examiner's action and did not reply to every ground of objection and rejection in the prior Office action. A general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references does not comply with the requirements of this section." (instant Office Action, page 3 lines 12-18).

Appellants respectfully submit that the instant response and the response to the Office Action mailed August 13, 2008 directed arguments to the grounds of rejections and specifically points out how the language of the claims is patentably distinguished from Bowman-Amuah. As argued above, Bowman-Amuah at least fails to disclose "a recovery agent for taking an action upon an occurrence of an exception that occurred for a programming task, wherein the action is performed outside of a debugging operation" as is disclosed by Appellants' Claim 1.

Additionally, Appellants respectfully submit that "an exception that occurred for a programming task," as disclosed by Appellants' Claim 1 should not be equated with "Batch jobs exceptions," as is argued in the instant Office Action. Appellants respectfully point out, as argued above, that Bowman-Amuah does not "batch jobs exceptions" nor does it disclose the handling of batch jobs exceptions. As point out above, Bowman-Amuah does disclose “batch processing” “used to perform large scale repetitive processing” and “batch application programs” including “report generation,” and does describe a type of “exception handling” in a different portion of Bowman-Amuah. However, Bowman-Amuah does not disclose “an exception handler for recording exception information dependant on types of exceptions and programming tasks that encounter exceptions” nor “a recovery agent for taking an action upon an occurrence of an exception that occurred for a programming task, wherein the action is performed outside of a debugging operation” as is disclosed by Appellants’ Claim 1.

Conclusion

Appellants believe that pending Claims 1-8, 10-15 and 17 are not anticipated by Bowman-Amuah and therefore overcome the rejection under 35 U.S.C. §102(b).

Accordingly, Appellants respectfully submit that the rejection of Claims 1-8, 10-15 and 17 under 35 U.S.C. §102(b) is improper and should be reversed.

The Appellants wish to encourage the Examiner or a member of the Board of Patent Appeals to telephone the Appellants' undersigned representative if it is felt that a telephone conference could expedite prosecution.

Respectfully submitted,
WAGNER BLECHER LLP

Dated: 05/27/2009

/John P. Wagner, Jr./
John P. Wagner, Jr.
Registration No. 35,398
123 Westridge Drive
Watsonville, CA 95076
(408) 377-0500

VIII. Appendix - Clean Copy of Claims on Appeal

1. An exception handling mechanism stored in one or more computer-readable storage devices, said exception handling mechanism comprising:

an exception handler for recording exception information dependant on types of exceptions and programming tasks that encounter
exceptions; and

a recovery agent for taking an action upon an occurrence of an exception that occurred for a programming task, wherein the action is performed outside of a debugging operation;

wherein the action to be taken upon the occurrence of the exception corresponds to a type of exception and a programming task, and includes one or a combination of restarting the programming task, terminating the programming task, resetting a system running the programming task, and disregarding the exception,

wherein the exception handler and the recovery agent run on a first system that operates autonomously and the first system is embedded in a second system.

2. The mechanism of claim 1 wherein the recorded exception information associated with an exception is associated with a signature for identifying the recorded exception information with its associated exception.

3. The mechanism of claim 2 wherein the signature includes a version of a program running the programming task.

4. The mechanism of claim 1 wherein a plurality of sets of exception information for a plurality of exceptions is maintained in the system running the programming task; each set of exception information being associated with a signature for identifying that set of exception information.

5. The mechanism of claim 1 wherein the recorded exception information associated with an exception is associated with a signature for identifying the format of the exception information.

6. The mechanism of claim 1 wherein the recorded exception information includes data related to a program stack, including data to reconstruct the program stack at time of exception.

7. The mechanism of claim 1 further comprising an analysis tool communicating via an interface with the system running the programming task, for identifying causes of the exception.

8. The mechanism of claim 7 wherein the analysis tool uses a version to match the object code of a program running the programming task to the source code of the program.

10. A processing system stored in one or more computer-readable storage devices, said processing system comprising:

a first system;

an autonomous second system embedded in the first system;

an exception handler running in the second system for recording exception information upon an occurrence of an exception in the second system; and

a recovery agent running on the second system, for taking an action upon the occurrence of the exception based on the recorded exception information, wherein the action is performed outside of a debugging operation;

wherein the action corresponds to a type of exception that occurred in a programming task.

11. The processing system of claim 10 further comprising an analysis tool for receiving, via an interface, the recorded exception information from the second system and for identifying the cause of the exception.

12. The processing system of claim 10 wherein the second system includes nonvolatile memory for storing exception information.

13. The processing system of claim 12 wherein the exception information stored in the non-volatile memory is compressed.

14. The processing system of claim 12 wherein the exception information stored in non-volatile memory includes a plurality of sets of exception information, each set being associated with an exception and a signature.

15. A computer-readable storage device having stored thereon a computing system comprising:

an exception handler for recording exception information on non-volatile memory upon an occurrence of an exception;

a recovery agent for taking an action upon the occurrence of the exception based on the recorded exception information, wherein the action is performed outside of a debugging operation; and

an analysis tool for identifying the cause of the exception;

wherein the analysis tool receives the exception information from the nonvolatile memory via an interface interfacing a first system and a second system running the exception handler and the recovery agent wherein the second system is embedded in a third system and the second system operates autonomously of other systems.

17. The computer-readable storage device of claim 15 wherein the recorded exception information includes data related to a program stack.

IX. Evidence Appendix

No evidence is herein appended.

X. Related Proceedings Appendix

No related proceedings.